



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,786	02/24/2004	Moshe E. Matsa	POU920030086US1	3750

23334 7590 05/02/2008
FLEIT, KAIN, GIBBONS, GUTMAN, BONGINI
& BIANCO P.L.
ONE BOCA COMMERCE CENTER
551 NORTHWEST 77TH STREET, SUITE 111
BOCA RATON, FL 33487

EXAMINER

LOVEL, KIMBERLY M

ART UNIT	PAPER NUMBER
----------	--------------

2167

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

05/02/2008

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptoboca@focusnip.com

DETAILED ACTION

Response to Amendment

1. This communication is in response to the Amendment filed 28 January 2008.
2. Claims 1, 2, 4-10 and 13-24 are currently pending. In the Amendment filed 28 January 2008, claims 1, 9 and 17 were amended; claims 3, 11 and 12 were canceled; and claims 22-24 were added. This action is made Final.
3. The prior art rejections made in the previous Office Action are maintained for the reasons discussed below.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. **Claims 1, 9, 17 and 24** are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The examiner fails to find antecedent basis for the steps of **directly** registering, **directly** notifying and nested configuration values.

35 USC § 101 Clarifications

6. Claims 9 and 17 are directed towards a program product and a system, respectively. The claims include a storage medium which is construed as representing the statutory subject matter defined as the non-volatile memory on page 25, lines 3-14 of the Applicants' Specification.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1, 2, 3-10 and 13-24 are rejected under 35 U.S.C. 102(b) as being anticipated by US PGPub 2001/0034771 to Hutsch et al (hereafter Hutsch).

Referring to claim 1, Hutsch discloses a method for managing configuration data, the method comprising the steps of:

storing a plurality of configuration values in a hierarchical tree [DOM tree] having a plurality of nodes (see [0327], lines 3-5), a defined structure [strongly typed schema] (see [0418]), and defined data types for the stored configuration values (see [0329]), wherein the plurality of nodes includes at least one inner node [i.e., color] and at least one child node [i.e., red] that is associated with the inner node (see Fig 16A), wherein each node is associated with at least one of the configuration values (see [0158]), and each of the configuration values dictates how an application component associated with

that configuration value at least one of behaves and interacts with other application components, and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values [value] and an identifier [key] associated with at least one application component (see [0158] and [0159]);

registering at least one application component with at least one of the nodes of the tree, based on at least one query [transaction] received from the at least one application component (see [0159]); and

directly notifying the at least one application component [listener] when a configuration value associated with the at least one node is modified [alterations], based on an addition or change in at least one configuration value that matches the at least one query [transaction] (see [0159]).

Referring to claim 2, Hutsch discloses the method of claim 1, wherein the at least one query depends on at least one of a location of a configuration value in the tree [value] and a data type of a configuration value (see [0159]).

Referring to claim 4, Hutsch discloses the method of claim 1, wherein the at least one application component comprises a plurality of components of an email application (see [0210] and [0316]).

Referring to claim 5, Hutsch discloses the method of claim 1, wherein a node further includes a reference to at least one node (see Fig 16A).

Referring to claim 6, Hutsch discloses the method of claim 1, wherein the notifying step comprises: modifying at least one configuration value that is associated

with the at least one node with which the at least one application component is registered; storing in the hierarchical tree the configuration value that was modified, the configuration value being stored at the at least one node with which the at least one application component is registered; and notifying the at least one application component that the configuration value was modified (see [0159]).

Referring to claim 7, Hutsch discloses the method of claim 6, further comprising the step of supplying the configuration value that was modified to the at least one application component (see [0159]).

Referring to claim 8, Hutsch discloses the method of claim 1, further comprising the step of supplying at least one of the configuration values stored in the hierarchical tree to the at least one application component (see [0159]).

Referring to claim 9, Hutsch discloses a computer program product for managing configuration data, the computer program product comprising:

a storage medium readable by a processing circuit and storing instructions for execution by the processing circuit [0683] for performing a method comprising the steps of:

storing a plurality of configuration values in a hierarchical tree [DOM tree] having a plurality of nodes (see [0327], lines 3-5), a defined structure [strongly typed schema] (see [0418]), and defined data types for the stored configuration values (see [0329]), wherein the plurality of nodes includes at least one inner node [i.e., color] and at least one child node [i.e., red] that is associated with the inner node (see Fig 16A), wherein each node is associated with at least one of the configuration values (see [0158]), and

each of the configuration values dictates how an application component associated with that configuration value at least one of behaves and interacts with other application components, and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values [value] and an identifier [key] associated with at least one application component (see [0158] and [0159]);

registering at least one application component directly with at least one of the nodes of the tree, based on at least one query [transaction] received from the at least one application component (see [0159]); and

directly notifying the at least one application component [listener] when a configuration value associated with the at least one node is modified [alterations], based on an addition or change in at least one configuration value that matches the at least one query [transaction] (see [0159]).

Referring to claim 10, Hutsch discloses the computer program product of claim 9, wherein the at least one query depends on at least one of a location of a configuration value in the tree [value] and a data type of a configuration value (see [0159]).

Referring to claim 13, Hutsch discloses the computer program product of claim 9, wherein a node further includes a reference to at least one node (see Fig 16A).

Referring to claim 14, Hutsch discloses the computer program product of claim 9, wherein the notifying step comprises: modifying at least one configuration value that is associated with the at least one node with which the at least one application component is registered; storing in the hierarchical tree the configuration value that was modified,

the configuration value being stored at the at least one node with which the at least one application component is registered; and notifying the at least one application component that the configuration value was modified (see [0159]).

Referring to claim 15, Hutsch discloses the computer program product of claim 14, wherein the method further comprises the step of supplying the configuration value that was modified to the at least one application component (see [0159]).

Referring to claim 16, Hutsch discloses the computer program product of claim 9, wherein the method further comprises the step of supplying at least one of the configuration values stored in the hierarchical tree to the at least one application component (see [0159]).

Referring to claim 17, Hutsch discloses a computer system for managing configuration data, the computer system comprising:

an organization module (see [0683]) for organizing a plurality of configuration values in a hierarchical tree [DOM tree] having a plurality of nodes (see [0327], lines 3-5), a defined structure [strongly typed schema] (see [0418]), and defined data types for the stored configuration values (see [0329]), wherein the plurality of nodes includes at least one inner node [i.e., color] and at least one child node [i.e., red] that is associated with the inner node (see Fig 16A), wherein each node is associated with at least one of the configuration values (see [0158]), and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated

with a combination of a set of configuration values [value] and an identifier [key] associated with at least one application component (see [0158] and [0159]);

a storage medium (see [0342]) for storing the plurality of configuration values in the hierarchical tree, each of the configuration values dictates how an application component associated with that configuration value at least one of behaves and interacts with other application components (see [0158] and [0159]);

a registration module (see [0683]) registering at least one application component directly with at least one of the nodes of the tree, based on at least one query [transaction] received from the at least one application component (see [0159]); and

a notification module (see [0683]) for directly notifying the at least one application component [listener] when a configuration value associated with the at least one node is modified [alterations], based on an addition or change in at least one configuration value that matches the at least one query [transaction] (see [0159]).

Referring to claim 18, Hutsch discloses the computer system of claim 17, wherein the at least one query depends on at least one of a location of a configuration value in the tree [value] and a data type of a configuration value (see [0159]).

Referring to claim 19, Hutsch discloses the computer system of claim 17, wherein the hierarchical tree is an Extensible Markup Language (XML) tree, and an XML schema describes the structure of the XML tree and the data types that are stored (see [0158] and [0321]).

Referring to claim 20, Hutsch discloses the computer system of claim 17, wherein the at least one application component comprises a plurality of components of an email application (see [0210] and [0316]).

Referring to claim 21, Hutsch discloses the method of claim 1, wherein the plurality of configuration values in the hierarchical tree includes all of the configuration data values that are required by the at least one application component (see [0158] and [0159]).

Referring to claim 22, Hutsch discloses the method of claim 1, wherein the step of registering at least one application component comprises registering the at least one application component with the at least one inner node (see [0159] and Fig 16A).

Referring to claim 23, Hutsch discloses the method of claim 22, wherein the step of directly notifying the at least one application component comprises directly notifying the at least one application component when at least one configuration value associated with at least one of the inner node and the child node that is associated with the inner node is modified, based on an addition or change in the at least one configuration value (see [0159] and Fig 16A).

Referring to claim 24, Hutsch discloses the method of claim 1, wherein at least one configuration value in the plurality of configuration values that is associated with a first application component overlaps with another configuration value in the plurality of configuration values that is associated with a second application component, and the at least one configuration value and the other configuration value are nested under a common sub-tree in the tree (see [0159]).

Response to Arguments

9. Applicant's arguments filed 28 January 2008 have been fully considered but they are not persuasive.

10. On pages 10-11 of the Remarks, the Applicants' argue the following: "Hutsch explicitly teaches that data only exists at leaf nodes of a tree. In contrast, in embodiments of the present invention configuration values exist at all nodes, not just leaf nodes."

The examiner respectfully disagrees. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., configuration values exist at all nodes) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). The claim language states "wherein each node is associated with at least one of the configuration values." When given the broadest reasonable interpretation, the concept of associating each node with a configuration value is not considered to restrict the interpretation of the claim limitation to mean that configuration values exist at all nodes. The configuration values of Hutsch are stored in a tree. The tree consists of parent-child relationships. Therefore, even if the values are only stored in the leaf nodes, the nodes located above the leaf node in the hierarchy have an association with the values in the leaf node.

Furthermore, page 8, lines 14-16 of the Applicants' specification states that "In one embodiment of the present invention, each node of the tree includes at least one

configuration value.” Therefore, since this is just one embodiment, the specification does not explicitly limit the term “associated” to mean that “a configuration value exists at all nodes.”

11. On page 11 of the Remarks, the Applicants’ argue the following: “Hutsch further teaches that listeners can register to be notified when alterations are made to configuration service elements. These listeners then notify the applications affected. Because data is only stored at leaf nodes, the listeners or components can only register an interest for data in a leaf node. ... In contrast, in embodiments of the present invention the application directly registers with the node itself. Stated differently, the reference to an application component that is to be notified when a change occurs is stored within the node itself.

The examiner respectfully disagrees that the concept of an application component registering as a listener fails to meet this limitation when given the broadest reasonable interpretation of the claimed limitation. The claim limitation is not considered to be explicitly limited to “storing a reference in the node itself” and therefore the Applicants’ interpretation of the claim language is not considered to be the only possible interpretation.

12. On page 12 of the Remarks, the Applicants’ argue that Hutsch does not disclose the limitation “the nodes include at least one inner node and at least one child node that is associated with the inner node.

The examiner respectfully disagrees. This is a newly added claim and is taught for the reasons cited above in the rejection.

13. On page 12 of the Remarks, the Applicants' argue the following: "However, there is support whatsoever in Hutsch for equating that the "key" of Hutsch is the same as the recited "identifier associated with at least one application component." Nowhere does Hutch teach or suggest that the "key" is an identifier associated with at least one application component."

The examiner respectfully disagrees. Since the key is associated with the node and the application is associated with the node, the key is considered to be associated with the application.

14. On page 12 of the Remarks, the Applicants' argue the following: "Furthermore, the claims recite that: 1) some of the nodes are only associated with a set of configuration values, and 2) other of the nodes are associated with a combination of a set of configuration values and an identifier associated with at least one application component."

The examiner respectfully disagrees. Paragraph [0159] of Hutsch that subtrees exist. Subtrees are considered to be associated with a plurality of configuration values and identifiers.

15. On page 13 of the Remarks, the Applicants' argue that Hutsch teaches away from the concept of an application component directly registering with at least one of the nodes of the tree. However, the examiner fails to find support in the Applicants' specification for this limitation and the Applicants failed to point out where support is found within the specification for the newly added limitation.

16. On pages 14-15 of the Remarks, the Applicants' argue that Hutsch teaches away from the concept of a node directly notifying an application component directly. However, the examiner fails to find support in the Applicants' specification for this limitation and the Applicants failed to point out where support is found within the specification for the newly added limitation.

17. The rejections of claims 2, 4-8, 10, 13-16, 18-24, which are dependent on claims 1, 9 and 17 respectively are maintained for the reasons stated above.

Conclusion

18. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KIMBERLY LOVEL whose telephone number is (571)272-2750. The examiner can normally be reached on 8:00 - 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/John R. Cottingham/
Supervisory Patent Examiner, Art Unit 2167

Kimberly Lovel
Examiner
Art Unit 2167

25 April 2008
kml